7. Linear equations

- QR factorization method
- factor and solve
- LU factorization
QR factorization and matrix inverse

QR factorization of nonsingular matrix

every nonsingular \( A \in \mathbb{R}^{n \times n} \) has a QR factorization

\[
A = QR
\]

- \( Q \in \mathbb{R}^{n \times n} \) is orthogonal \( (Q^T Q = QQ^T = I) \)
- \( R \in \mathbb{R}^{n \times n} \) is upper triangular with positive diagonal elements

Inverse from QR factorization: the inverse \( A^{-1} \) can be written as

\[
A^{-1} = (QR)^{-1} = R^{-1} Q^{-1} = R^{-1} Q^T
\]
Solving linear equations by QR factorization

**Algorithm:** to solve $Ax = b$ with nonsingular $A \in \mathbb{R}^{n \times n}$,

1. factor $A$ as $A = QR$
2. compute $y = Q^T b$
3. solve $Rx = y$ by back substitution

**Complexity:** $2n^3 + 3n^2 \approx 2n^3$ flops

- QR factorization: $2n^3$
- matrix-vector multiplication: $2n^2$
- back substitution: $n^2$
Multiple right-hand sides

consider $k$ sets of linear equations with the same coefficient matrix $A$:

$$Ax_1 = b_1, \quad Ax_2 = b_2, \quad \ldots, \quad Ax_k = b_k$$

• equivalently, solve $AX = B$ where $B$ is $n \times k$ matrix with columns $b_1, \ldots, b_k$
• can be solved in $2n^3 + 3kn^2$ flops if we reuse the factorization $A = QR$
• for $k \ll n$, cost is roughly equal to cost of solving one equation ($2n^3$)

Application: to compute $A^{-1}$, solve the matrix equation $AX = I$

• equivalent to $n$ equations

$$Rx_1 = Q^T e_1, \quad Rx_2 = Q^T e_2, \quad \ldots, \quad Rx_n = Q^T e_n$$

($x_i$ is column $i$ of $X$ and $Q^T e_i$ is transpose of $i$th row of $Q$)
• complexity is $2n^3 + n^3 = 3n^3$ (here the 2nd term $n^3$ is not negligible)
Outline

- QR factorization method
- factor and solve
- LU factorization
Factor–solve approach

to solve $Ax = b$, first write $A$ as a product of “simple” matrices

$$A = A_1 A_2 \cdots A_k$$

then solve $(A_1 A_2 \cdots A_k)x = b$ by solving $k$ equations

$$A_1 z_1 = b, \quad A_2 z_2 = z_1, \quad \ldots, \quad A_{k-1} z_{k-1} = z_{k-2}, \quad A_k x = z_{k-1}$$

Examples

• QR factorization: $k = 2$, $A = QR$

• LU factorization (this lecture)

• Cholesky factorization (later)
Complexity of factor–solve method

#flops = \( f + s \)

- \( f \) is complexity of factoring \( A \) as \( A = A_1A_2\cdots A_k \) (factorization step)
- \( s \) is complexity of solving the \( k \) equations for \( z_1, z_2, \ldots z_{k-1}, x \) (solve step)
- usually \( f \gg s \)

**Example:** solving linear equations using the QR factorization

\[
\begin{align*}
f &= 2n^3, \\
s &= 3n^2
\end{align*}
\]
Outline

- QR factorization method
- factor and solve
- LU factorization
LU factorization

LU factorization without pivoting

\[ A = LU \]

- \( L \) unit lower triangular, \( U \) upper triangular
- does not always exist (even if \( A \) is nonsingular)

LU factorization (with row pivoting)

\[ A = PLU \]

- \( P \) permutation matrix, \( L \) unit lower triangular, \( U \) upper triangular
- exists if and only if \( A \) is nonsingular (see later)

**Complexity:** \((2/3)n^3\) if \( A \) is \( n \times n \)
Solving linear equations by LU factorization

**Algorithm:** to solve $Ax = b$ with nonsingular $A$ of size $n \times n$

1. factor $A$ as $A = PLU$ ($(2/3)n^3$ flops)
2. solve $(PLU)x = b$ in three steps
   
   (a) permutation: $z_1 = P^T b$ (0 flops)
   (b) forward substitution: solve $Lz_2 = z_1$ ($n^2$ flops)
   (c) back substitution: solve $Ux = z_2$ ($n^2$ flops)

**Complexity:** $(2/3)n^3 + 2n^2 \approx (2/3)n^3$ flops

this is the standard method for solving $Ax = b$
Multiple right-hand sides

Two equations with the same matrix $A$ (nonsingular and $n \times n$):

$$Ax = b, \quad A\tilde{x} = \tilde{b}$$

- factor $A$ once
- forward/back substitution to get $x$
- forward/back substitution to get $\tilde{x}$

Complexity: $(2/3)n^3 + 4n^2 \approx (2/3)n^3$

Exercise: propose an efficient method for solving

$$Ax = b, \quad A^T\tilde{x} = \tilde{b}$$
LU factorization and matrix inverse

suppose $A$ is nonsingular and $n \times n$, with LU factorization

$$A = PLU$$

- inverse from LU factorization

$$A^{-1} = (PLU)^{-1} = U^{-1}L^{-1}P^T$$

- gives interpretation of solve step: we evaluate

$$x = A^{-1}b = U^{-1}L^{-1}P^Tb$$

in three steps

$$z_1 = P^Tb, \quad z_2 = L^{-1}z_1, \quad x = U^{-1}z_2$$
Computing the inverse

solve $AX = I$ column by column

- one LU factorization of $A$: $2n^3/3$ flops
- $n$ solve steps: $2n^3$ flops
- total: $(8/3)n^3$ flops

slightly faster methods exist that exploit structure in right-hand side $I$

**Conclusion**: do not solve $Ax = b$ by multiplying $A^{-1}$ with $b$
LU factorization without pivoting

\[
\begin{bmatrix}
A_{11} & A_{1,2:n} \\
A_{2:n,1} & A_{2:n,2:n}
\end{bmatrix} =
\begin{bmatrix}
1 & 0 \\
L_{2:n,1} & L_{2:n,2:n}
\end{bmatrix}
\begin{bmatrix}
U_{11} & U_{1,2:n} \\
0 & U_{2:n,2:n}
\end{bmatrix}
= \begin{bmatrix}
U_{11} & U_{1,2:n} \\
U_{11}L_{2:n,1} & L_{2:n,1}U_{1,2:n} + L_{2:n,2:n}U_{2:n,2:n}
\end{bmatrix}
\]

Recursive algorithm

- determine first row of \( U \) and first column of \( L \)

\[
U_{11} = A_{11}, \quad U_{1,2:n} = A_{1,2:n}, \quad L_{2:n,1} = \frac{1}{A_{11}}A_{2:n,1}
\]

- factor the \((n - 1) \times (n - 1)\)-matrix \( A_{2:n,2:n} - L_{2:n,1}U_{1,2:n} \) as

\[
A_{2:n,2:n} - L_{2:n,1}U_{1,2:n} = L_{2:n,2:n}U_{2:n,2:n}
\]

this is an LU factorization (without pivoting) of size \((n - 1) \times (n - 1)\)
Example

LU factorization (without pivoting) of

\[ A = \begin{bmatrix} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{bmatrix} \]

write as \( A = LU \) with \( L \) unit lower triangular, \( U \) upper triangular

\[ A = \begin{bmatrix} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 6 & 7 & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \]
Example

- First row of $U$, first column of $L$:

\[
\begin{bmatrix}
8 & 2 & 9 \\
4 & 9 & 4 \\
6 & 7 & 9
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
1/2 & 1 & 0 \\
3/4 & L_{32} & 1
\end{bmatrix}\begin{bmatrix}
8 & 2 & 9 \\
0 & U_{22} & U_{23} \\
0 & 0 & U_{33}
\end{bmatrix}
\]

- Second row of $U$, second column of $L$:

\[
\begin{bmatrix}
9 & 4 \\
7 & 9
\end{bmatrix}
- \begin{bmatrix}
1/2 \\
3/4
\end{bmatrix}\begin{bmatrix}
2 & 9
\end{bmatrix}
= \begin{bmatrix}
1 & 0 \\
L_{32} & 1
\end{bmatrix}\begin{bmatrix}
U_{22} & U_{23} \\
0 & U_{33}
\end{bmatrix}
\]

\[
\begin{bmatrix}
8 & -1/2 \\
11/2 & 9/4
\end{bmatrix}
= \begin{bmatrix}
1 & 0 \\
11/16 & 1
\end{bmatrix}\begin{bmatrix}
8 & -1/2 \\
0 & U_{33}
\end{bmatrix}
\]

- Third row of $U$: $U_{33} = 9/4 + 11/32 = 83/32$

Conclusion

\[
A = \begin{bmatrix}
8 & 2 & 9 \\
4 & 9 & 4 \\
6 & 7 & 9
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
1/2 & 1 & 0 \\
3/4 & 11/16 & 1
\end{bmatrix}\begin{bmatrix}
8 & 2 & 9 \\
0 & 8 & -1/2 \\
0 & 0 & 83/32
\end{bmatrix}
\]

Linear equations 7.14
Not every nonsingular $A$ can be factored as $A = LU$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- first row of $U$, first column of $L$:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & L_{32} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- second row of $U$, second column of $L$:

$$\begin{bmatrix} 0 & 2 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ L_{32} & 1 \end{bmatrix} \begin{bmatrix} U_{22} & U_{23} \\ 0 & U_{33} \end{bmatrix}$$

$U_{22} = 0$, $U_{23} = 2$, $L_{32} \cdot 0 = 1$?
LU factorization (with row pivoting)

if \( A \) is \( n \times n \) and nonsingular, then it can be factored as

\[
A = PLU
\]

\( P \) is a permutation matrix, \( L \) is unit lower triangular, \( U \) is upper triangular

- not unique; there may be several possible choices for \( P, L, U \)
- interpretation: permute the rows of \( A \) and factor \( P^TA \) as \( P^TA = LU \)
- also known as *Gaussian elimination with partial pivoting* (GEPP)
- complexity: \((2/3)n^3\) flops

we skip the details of calculating \( P, L, U \)
Example

\[
\begin{bmatrix}
0 & 5 & 5 \\
2 & 9 & 0 \\
6 & 8 & 8
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0
\end{bmatrix}\begin{bmatrix}
1 & 0 & 0 \\
1/3 & 1 & 0 \\
0 & 15/19 & 1
\end{bmatrix}\begin{bmatrix}
6 & 8 & 8 \\
0 & 19/3 & -8/3 \\
0 & 0 & 135/19
\end{bmatrix}
\]

the factorization is not unique; the same matrix can be factored as

\[
\begin{bmatrix}
0 & 5 & 5 \\
2 & 9 & 0 \\
6 & 8 & 8
\end{bmatrix}
= \begin{bmatrix}
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
3 & -19/5 & 1
\end{bmatrix}\begin{bmatrix}
2 & 9 & 0 \\
0 & 5 & 5 \\
0 & 0 & 27
\end{bmatrix}
\]
Effect of rounding error

\[
\begin{bmatrix}
10^{-5} & 1 \\
1 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= \begin{bmatrix}
1 \\
0
\end{bmatrix}
\]

solution:

\[
x_1 = \frac{-1}{1 - 10^{-5}}, \quad x_2 = \frac{1}{1 - 10^{-5}}
\]

- let us solve using LU factorization for the two possible permutations:

\[
P = \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix} \quad \text{or} \quad P = \begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}
\]

- we round intermediate results to four significant decimal digits
First choice: \( P = I \) (no pivoting)

\[
\begin{bmatrix}
10^{-5} & 1 \\
1 & 1
\end{bmatrix} = 
\begin{bmatrix}
1 & 0 \\
10^5 & 1
\end{bmatrix} 
\begin{bmatrix}
10^{-5} & 1 \\
0 & 1 - 10^5
\end{bmatrix}
\]

- \( L, U \) rounded to 4 significant decimal digits

\[
L = 
\begin{bmatrix}
1 & 0 \\
10^5 & 1
\end{bmatrix}, \quad U = 
\begin{bmatrix}
10^{-5} & 1 \\
0 & -10^5
\end{bmatrix}
\]

- forward substitution

\[
\begin{bmatrix}
1 & 0 \\
10^5 & 1
\end{bmatrix} 
\begin{bmatrix}
z_1 \\
z_2
\end{bmatrix} = 
\begin{bmatrix}
1 \\
0
\end{bmatrix} \quad \Rightarrow \quad z_1 = 1, \quad z_2 = -10^5
\]

- back substitution

\[
\begin{bmatrix}
10^{-5} & 1 \\
0 & -10^5
\end{bmatrix} 
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} = 
\begin{bmatrix}
1 \\
-10^5
\end{bmatrix} \quad \Rightarrow \quad x_1 = 0, \quad x_2 = 1
\]

error in \( x_1 \) is 100%
Second choice: interchange rows

\[
\begin{bmatrix}
1 & 1 \\
10^{-5} & 1
\end{bmatrix}
= \begin{bmatrix}
1 & 0 \\
10^{-5} & 1
\end{bmatrix}
\begin{bmatrix}
1 & 1 \\
0 & 1 - 10^{-5}
\end{bmatrix}
\]

- \(L, U\) rounded to 4 significant decimal digits

\[
L = \begin{bmatrix}
1 & 0 \\
10^{-5} & 1
\end{bmatrix}, \quad U = \begin{bmatrix}
1 & 1 \\
0 & 1
\end{bmatrix}
\]

- forward substitution

\[
\begin{bmatrix}
1 & 0 \\
10^{-5} & 1
\end{bmatrix}
\begin{bmatrix}
z_1 \\
z_2
\end{bmatrix}
= \begin{bmatrix}
0 \\
1
\end{bmatrix}
\implies z_1 = 0, \quad z_2 = 1
\]

- backward substitution

\[
\begin{bmatrix}
1 & 1 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= \begin{bmatrix}
0 \\
1
\end{bmatrix}
\implies x_1 = -1, \quad x_2 = 1
\]

error in \(x_1, x_2\) is about \(10^{-5}\)
Conclusion: rounding error and LU factorization

- for some choices of $P$, small errors in the algorithm can cause very large errors in the solution
- this is called *numerical instability*: for the first choice of $P$ in the example, the algorithm is unstable; for the second choice of $P$, it is stable
- from numerical analysis: there is a simple rule for selecting a good permutation
  (we skip the details, since we skipped the details of the factorization)
Sparse linear equations

if $A$ is sparse, it is usually factored as

$$A = P_1LU P_2$$

$P_1$ and $P_2$ are permutation matrices

- interpretation: permute rows and columns of $A$ and factor $\tilde{A} = P_1^T A P_2^T$

$$\tilde{A} = LU$$

- choice of $P_1$ and $P_2$ greatly affects the sparsity of $L$ and $U$: several heuristic methods exist for selecting good permutations

- in practice: $\#\text{flops} \ll (2/3)n^3$; exact value depends on $n$, number of nonzero elements, sparsity pattern
Conclusion

different levels of detail in understanding how linear equation solvers work

**Highest level**

- $x = A \backslash b$ costs $(2/3)n^3$
- more efficient than $x = \text{inv}(A) \ast b$

**Intermediate level:** factorization step $A = PLU$ followed by solve step

**Lowest level:** details of factorization $A = PLU$

- for most applications, level 1 is sufficient
- in some situations (e.g., multiple right-hand sides) level 2 is useful
- level 3 is important for experts who write numerical libraries